SemEval 2024 Task 8: Machine-Generated Text Detection

Xiao Yang[*] and Mingze Li[†]

[*]Department of Mathematics and Statistics, Carleton University

Abstract

This project is the subtask A of SemEval-2024 Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection. Given a full text, we determine whether it is human-written or machine-generated (Jawahar et al., 2020). We restrict ourselves to sub-task A: monolingual, only the English texts. We provide a comparative study of traditional classifiers from machine learning via recent LLMs to demonstrate the classification task performance.

1 Introduction

Large language models (LLMs) are becoming mainstream, leading to an explosion of machine-generated content over different fields, such as news, social media, question-answering forums, educational, and even academic contexts (Jawahar et al., 2020). Recent LLMs, such as Chat GPT and GPT-4, generate remarkably fluent responses to a wide variety of user queries. The articulate nature of such generated texts makes LLMs attractive to replace human labor in many scenarios. However, this has also resulted in concerns regarding their potential misuse, such as spreading misinformation in the education system. Since humans perform only slightly better than the models when classifying machine-generated vs. human-written text, there is a need to develop an automatic system to identify machine-generated text with the goal of mitigating potential misuse.

Although there are much mature systems for complex classification task developed (Sun et al., 2023), we develop such an easier automatic system that takes a data frame of human-written and machine generated texts to assign a label to each snippet of texts for identification. 0 stands for human-written texts, while 1 means the text is generated by machine.

[*]barnettyang@cmail.carleton.ca

[†]MingzeLi7@cmail.carleton.ca

2 Data for Shared Task

The primary data is provided by (Jawahar et al., 2020), it can be accessed in https://drive.google.com/drive/folders/1CAbb3DjrOPBNm0ozVBfhvrEh9P9rAppc, which contains 119,757 texts as monolingual training data. The test data can be found at https://drive.google.com/drive/folders/10DKtClzkwIIAatzHBWXZXuQNID-DNGSG. They are described in (Wang et al., 2023). The data format is in .json. The output labels, along with text ids form a numpy array of dimension (number of text ids $\times 2$), which are converted to a separate .json file. The scorer script outputs the metrics, including accuracy, micro-F1, and macro-F1 scores.

3 Methodology

In terms of text categorization task, there are many existing classifiers from classical statistics. We started with the classic support vector machine (SVM) with a linear kernel, then use the Gaussian kernel for more non-linearity. Next, we use a default Naive Bayes.

3.1 SVM

Consider our binary classification task have two labels $\{1, -1\}$, the classical SVM with linear kernel formulates as follow. Consider a slack variable ξ_i when $C \neq 0$, the objective tries to minimize ξ_i, so we have the unconstrained mathematical formulation of SVM (James et al., 2013).

$$\xi_i = \begin{cases} 1 - y_i(\mathbf{w}^T\mathbf{x} + \mathbf{b}), & y_i(\mathbf{w}^T\mathbf{x} + \mathbf{b}) < 1, \\ 0, & y_i(\mathbf{w}^T\mathbf{x} + \mathbf{b}) > 1, \end{cases}$$

which is equivalent to

$$\xi = \max(1 - y_i(\mathbf{w}^T\mathbf{x} + \mathbf{b}), 0) \tag{1}$$

we obtain the unconstrained version of optimization problem, with L_2 regularization term and

hinge loss term, where C is a constant.

$$\min_{\mathbf{w},\mathbf{b}} \mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n} \max(1 - y_i(\mathbf{w}^T\mathbf{x} + \mathbf{b}), 0) \quad (2)$$

Hence, 1 gives the standard SVM with linear kernel. We next add more non-linearity with a Gaussian kernel. The decision function is given by (James et al., 2013)

$$h(z) = \text{sign}\left(\sum_{i=1}^{n} y_i\alpha_i k(\mathbf{x_i}, \mathbf{z}) + b\right) \quad (3)$$

where $0 < \alpha_i < C$ are constants, $k(\mathbf{x_i}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|)/\sigma$, $\sigma > 0$. The implementation is through default SVM function in scikit-learn package (Pedregosa et al., 2011), the whole training process took around three hours for linear kernel SVM and four hours for Gaussian kernel SVM on a standard CPU.

3.2 Naive Bayes

Let X and Y be two independent random variables, where $X = (X_1, X_2, \ldots X_d)$ represents the input data, and X_is are independent features. $Y \in \{y_1, y_2, \ldots y_k\}$, y_k is a positive integer representing class, assuming there are k classes. We aim to complete a classification task by finding a classifier (function) $f : X \to Y$, equivalently a probability $P(Y|X)$. Further assume $\mathbf{x} = (x_1, x_2, \ldots x_d)$ and y_i are realizations (samples) of X and Y, by Bayes rule, we have (Mitchell, 1997)

$$\begin{aligned} \hat{y_i} &= \text{argmax}_{y_i} P(Y = y_i | X_1, X_2 \ldots X_d) \\ &= \text{argmax}_{y_i} \prod_{i=1}^{d} P(X_i|Y = y_i)P(Y = y_i) \end{aligned}$$

$\hat{y_i} \in \{0, 1\}$, $P(Y = y_i)$ and $P(X_i|Y = y_i)$ are given by maximum likelihood estimators. The Bayesian classifier took only one hour to complete assigning labels to test data set.

3.3 Fine-Tune BERT

BERT (Devlin et al., 2019) is a baseline transformer model trained by Google in 2019, as a classic LLM, we fine tune the "bert-base-cased" version of this model (Devlin et al., 2018), which is available on hugging face. BERT is known as a general purpose LLM, whose novel approach of bidirectional representation of sentences and masked modelling improved old LSTM based models. The BERT framework mainly consists of pre-training on unlabeled data for different tasks and fine-tuning using labeled data from downstream tasks. Each downstream task has separate fine-tuned models, even though they are initialized with the same pre-trained parameters (Devlin et al., 2019). We refer to (Vaswani et al., 2017) for a detailed discussion of transformer model architecture.

In the "bert-base-cased" model, denote the number of layers (Transformer blocks) as L, hidden size as H, and the number of self-attention heads as A. "bert-base" has $L = 12$, $H = 768$, $A = 12$, with a total of 110M parameters (Devlin et al., 2019), same model size to compare with GPT. Following the baseline idea, we ran our fine-tuning via an A100 GPU on google colab, the training loop took roughly three hours.

	Accuracy	**Micro-F1**	**Macro-F1**
SVM_Linear	0.75607	0.75607	0.75557
SVM_Gaussian	0.78930	0.78930	0.78899
Naive_Bayes	0.83351	0.83351	0.83191
Logistic_Reg	0.76133	0.76196	0.76196
BERT	0.65196	0.65196	0.61459
RoBERTa	0.86397	0.86397	0.86316

Table 1: Evaluation metric scores of different classifiers on the monolingual test data set. See (Harbecke et al., 2022) for a discussion of F-scores.

3.4 Fine-Tune RoBERTa

A Robustly Optimized BERT Pretraining Approach, also called RoBERTa. Knowing that language model pre-train can lead to a significant performance improvement, a group of researchers reviewed all the details in the building process of BERT, and optimized hyperparameters in BERT. They also used an additional novel and CC-NEWS dataset. They found that BERT was significantly undertrained, and can match or exceed the performance of every model published after it. (Liu et al., 2019)

One of the distinct advantages of BERT and RoBERTa is the size. Different from other LLMs that needs 500MB to 15GB download, BERT needs only around 1MB, and RoBERTa 2.7MB. Some LLMs like gema and llama require GPU RAM of around 42 GB, which overshot our biggest available GPU(40GB). Therefore we cannot implement these models. The model we used is "roberta-base" model, it has a similar architecture as "bert-base-cased" model. The implementation of RoBERTa

also follows the baseline, and runs on the A100 GPU on Google Colab. The program also took around 3 hours.

4 Result

Our result for classification can be summarized in table 1. We can clearly see, for a large text data set, the Naive Bayes classifier performed the best in this binary classification task. The fine-tuned LLMs did not achieve a result as good as we expected. This indicates introducing new mechanisms such as in (Sun et al., 2023) can potentially improve the performance of LLMs. Also, we doubt the under-performance of BERT can potentially be a consequence of lack of computational power, as the fine tuning trainer only ran three epochs in total.

5 Conclusion and Future Work

We proposed a system that classify a given text into whether it is machine-generated or not. The comparative performance on the gold-labeled results are of noticeable results.

LLaMa (Touvron et al., 2023) is another recently proposed transformer model trained by Meta. It outperforms GPT-3 (Brown et al., 2020) in many benchmark tasks. LLaMa adopted a modified transformer architecture and training method from (Vaswani et al., 2017), but only used publicly available data sets to 'democratize' the usage of LLMs, as LLaMa can be run on a single GPU, and the model itself is free and open-source. Due to GPU's VRAM limit, we seek to use LLaMa if any RTX4090 becomes available.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

David Harbecke, Yuxuan Chen, Leonhard Hennig, and Christoph Alt. 2022. Why only micro-f1? class weighting of measures for relation classification. In *Proceedings of NLP Power! The First Workshop on Efficient Benchmarking in NLP*, pages 32–41, Dublin, Ireland. Association for Computational Linguistics.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning: with Applications in R*. Springer.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V. S. Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. *ArXiv*, abs/2011.01314.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Tom M Mitchell. 1997. *Machine learning*, volume 1. McGraw-hill New York.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *Conference on Empirical Methods in Natural Language Processing*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023.

M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *arXiv:2305.14902*.